REMARKS

Claims 1-19 are pending. In the Office Action mailed on November 7, 2005, the Examiner rejected claims 1-19 under 35 U.S.C. § 102(b) over U.S. Patent No. 5,542,058 to Brown III et al. ("Brown"). Applicants herein amend claims 1, 5, 10 and 16 to more particularly point out and distinctly claim Applicants' invention. Further examination and review in view of the amendments and remarks below are respectfully requested.

The Examiner indicated that Applicants' November 3, 2003, Information Disclosure Statement has been placed in the file, but is not being considered for failing to comply with the provisions of 37 C.F.R. §§ 1.97 and 1.98 and MPEP § 609 because only one of the non-patent literature references was included in the application. Even though copies of these non-patent literature references were cited in the parent case from which this application depends, Applicants herein resubmit as attachments copies of the non-patent literature references cited in its November 3, 2003 Information Disclosure Statement to address the Examiner's concern.

Applicants' techniques are directed to detecting when a portion of a computer program is improperly accessing a memory location. One aspect of Applicants' techniques provides a pointer in memory that is used by the computer program to access the memory location. The pointer contains an indication of whether traps to the pointed to memory location are enabled or disabled. The memory location also has a trap indicator, which is different from the aforementioned indication contained in the pointer, and which can be set to indicate that a trap should occur when the memory location is accessed. Then, when the computer program uses the pointer to access the memory location, the combination of the trap indication contained in the pointer and the trap indicator associated with the memory location determines whether a trap should or should not occur. For example, a trap occurs only if both the trap indication contained in the pointer and the trap indicator associated with the memory location are set to indicate that the trap should occur. Otherwise, the trap does not occur. This allows an authorized portion of a computer

program to access a protected memory location (i.e., a memory location whose trap indicator is set to indicate that a trap should occur when the memory location is accessed) without causing a trap by using a pointer that has its indication set to indicate that traps are disabled, and using this pointer to access the protected memory location. The trap indicator associated with the protected memory location does not need to be changed to indicate that the trap should not occur. In this scenario, even though the trap indicator associated with the protected memory location continues to indicate that the trap should occur, the trap does not occur because the indication contained in the pointer indicates that traps are disabled. In contrast, a portion of the program that is not authorized to access the protected memory location uses a pointer with the trap enabled. Thus, when the unauthorized portion uses the pointer to access the protected memory location (e.g., as a result of a bug in the program), a trap will occur and the access will be denied.

All of the claims stand rejected over Brown. Applicants respectfully traverse the Examiner's rejections. Brown merely describes a macropipelined microprocessor that sequentially buffers operands in queues during instruction decode, and then removes the queued instructions in order during instruction execution. (col. 5, lines 11-15.) A microinstruction control unit of the microprocessor contains a microtrap selector that allows for requesting a microtrap for dealing with abnormal events that require immediate service. The microtrap selector has a number of inputs for various conditions, and the microtrap selector applies an address to a selector, which feeds into the current address latch, under the specified conditions. When a microtrap occurs, the microcode control is transferred to the service microroutine beginning at this microtrap address. (Figure 12 and described at col. 27, line 25-col. 28, line 17.)

Claims 1-4 and 16-19 each recite: (1) setting a memory location to indicate a trap should occur when the memory location is accessed, and (2) under control of an authorized portion of a computer program, setting a pointer to point to the memory location, the pointer indicates that traps to the pointed to memory location are disabled, and accessing the memory location using the set pointer so that a trap does not occur and

access to the restricted memory location is allowed, or similar language. In rejecting the claims, the Examiner indicated that Brown's specifier queue synchronization counter corresponds to Applicants' setting a memory location to indicate a trap should occur when the memory location is accessed. In particular, the Examiner stated in pages 3 and 7 of the present Office Action that "the 'specifier queue synchronization counter' is what captures (traps) synchronization points to coordinate memory request operations."

Applicants respectfully disagree. Brown's specifier queue synchronization pointer is not an indicator that a trap should occur when the memory location is accessed. Rather, Brown's specifier queue synchronization counter is a buffer for synchronizing explicit memory requests across autonomous function units. (refer to detailed discussion of the operation of "spec-queue sync counter" at col. 49, line 35-col. 52, line 23.)

The Examiner also indicated that Brown's pointer to the data in a register file (col. 10, lines 22-32) and microtrap selector (col. 28, lines 44-47) correspond to Applicants' under control of an authorized portion of a computer program, setting a pointer to point to the memory location, the pointer indicates that traps to the pointed to memory location are disabled, and accessing the memory location using the set pointer so that a trap does not occur and access to the restricted memory location is allowed.

Applicants respectfully disagree. First, Brown neither teaches nor suggests that its pointer indicates that traps to the pointed to memory location are disabled, as recited in Applicants' claims. Second, Brown's microtrap selector applies an address to the current address latch based on a condition indicated on its inputs. (col. 28, lines 8-17; col. 27, lines 60-65.) Thus, in Brown, a trap does or does not occur based on the inputs of the microtrap selector. This is in contrast to (1) setting a memory location to indicate a trap should occur when the memory location is accessed, and (2) under control of an authorized portion of a computer program, setting a pointer to point to the memory location, the pointer indicates that traps to the pointed to memory location are disabled, and accessing the memory location using the set pointer so that a trap does not occur and

access to the restricted memory location is allowed, as recited. As described above, in Applicants' techniques, even though the memory location is set to indicate that a trap should occur when the memory location is accessed, the trap does not occur because the pointer that was used to access the memory location indicated that the traps are disabled. Moreover, the Examiner has failed to indicate where Brown discloses or suggests an authorized portion of a computer program: (1) setting a pointer to point to the memory location, where the pointer indicates that traps to the pointed to memory location are disabled, and (2) accessing the memory location using the set pointer so that a trap does not occur and access to the restricted memory location is allowed, as recited. Applicants can find in Brown no such disclosure or suggestion.

Claims 5-9 each recite (1) setting a pointer to a memory location wherein the pointer has an indication of trap handling depending on whether an unauthorized or authorized portion of the computer program is accessing the memory location, and (2) accessing the memory location and handling a trap wherein propriety of the access is determined based on the indication that the trap should occur when the memory location is accessed and the indication of trap handling in the pointer set to the memory location. Although Brown discloses a microtrap selector, Brown neither discloses nor suggests handling a trap based on the indication that the trap should occur when the memory location is accessed and the indication of trap handling in the pointer set to the memory location. As discussed above, in Brown, a trap does or does not occur based on the inputs of the microtrap selector. This is in contrast to (1) setting a pointer to a memory location wherein the pointer has an indication of trap handling depending on whether an unauthorized or authorized portion of the computer program is accessing the memory location, and (2) accessing the memory location and handling a trap wherein propriety of the access is determined based on the indication that the trap should occur when the memory location is accessed and the indication of trap handling in the pointer set to the memory location, as recited. Applicants can find in Brown no such disclosure or suggestion.

Claims 10-15 each recite (1) a pointer to an element in the data structure, the pointer having a first indication of whether a trap is enabled depending on whether an unauthorized or authorized portion of a computer program is accessing the data structure, and (2) for each element, a second indication of whether a trap is enabled, the second indication being distinct from the first indication. Again, Brown merely describes a microtrap selector for requesting a microtrap. (col. 28, lines 8-17.) This is in contrast to two distinct indications (i.e., the first indication associated with the pointer, and the second indication associated with each element in the data structure) for determining whether a trap is enabled.

In rejecting these claims, the Examiner indicated that Brown's source queue which holds a pointer to the data in the register file (col. 10, lines 22-32) corresponds to Applicants' for each element, a second indication of whether a trap is enabled. Previously, the Examiner stated that this pointer to the data in the register file (col. 10, lines 22-32) corresponded to Applicants' pointer to the memory location (e.g., the element in the data structure) in rejecting claims 1-4 and 16-19. Irregardless of whether Brown's pointer corresponds to Applicants' pointer to an element in the data structure, the pointer having a first indication of whether a trap is enabled depending on whether an unauthorized or authorized portion of a computer program is accessing the data structure, or Applicants' for each element, a second indication of whether a trap is enabled, Brown's pointer is merely a pointer to data and does not contain an indication of whether a trap is enabled. Applicants can find in Brown no such disclosure or suggestion.

Conclusion

In view of the foregoing, Applicants respectfully submit that claims 1-19 are allowable and ask that this application be passed to allowance. If the Examiner has any questions or believes a telephone conference would expedite prosecution of this application, the Examiner is encouraged to call the undersigned at (206) 359-8000.

Applicants believe no fee is due with this response. However, if a fee is due, please charge our Deposit Account No. 50-0665, under Order No. 324758003US8 from which the undersigned is authorized to draw.

Dated: January 24, 2006

Respectfully submitted,

By _____

Do Te Kim
Registration No.: 46,231
PERKINS COIE LLP
P.O. Box 1247
Seattle, Washington 98111-1247
(206) 359-8000
(206) 359-7198 (Fax)
Attorney for Applicants